

AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions and listings of claims in the application:

LISTING OF CLAIMS:

1. (currently amended): A flash memory access apparatus, comprising:

a flash memory comprising a plurality of units, each of the units comprising a plurality of blocks; and

a flash memory controller,

wherein if a write operation is requested for a logical block number of the flash memory, the flash memory controller is configured to write data and meta-information in a physical block corresponding to a logical block with the logical block number if a previous write operation has not been performed for the logical block, and the flash memory controller is configured to perform a write operation for writing the data and the meta-information allocated to the logical block in a new physical block without changing flash memory state information, wherein the flash memory state information is written in a previous physical block corresponding to the logical block if the previous write operation has been performed for the logical block, and

wherein the flash memory state information is time independent.

2. (original): The apparatus as claimed in claim 1, wherein the meta-information is written after the data of the logical block is written.

3. (original): The apparatus as claimed in claim 1, wherein the data and meta-information of the logical block are simultaneously written.

4. (previously presented): The apparatus as claimed in claim 1, wherein the meta-information comprises the logical block number, and the flash memory state information indicating a state of the physical block as valid, deleted, or invalid.

5. (previously presented): The apparatus as claimed in claim 1, wherein the flash memory controller is configured to perform a recovery operation which detects, during a scanning process, physical blocks for the logical block number and recovers from an error by determining a valid block for the logical block among the detected physical blocks.

6. (previously presented): The apparatus as claimed in claim 5, wherein the scanning process comprises reading a logical block number for each of the physical blocks by investigating the flash memory based on a latest accessed block, and investigating a field of a block allocation table corresponding to the read logical block number.

7. (previously presented): The apparatus as claimed in claim 6, wherein the investigating the field of the block allocation table comprises writing a state value of "1" in the field of the block allocation table if the state value has been "0," and detecting that the logical

block number has been searched for through the previous physical block during the scanning process, if the state value is "1."

8. (previously presented): The apparatus as claimed in claim 5, wherein the recovery operation recovers from an error by determining a latest accessed physical block for the logical block number among the detected physical blocks according to priorities set during the scanning process, as the valid block, and rewriting flash memory state information written in other physical blocks of the detected physical blocks as deleted.

9. (previously presented): The apparatus as claimed in claim 5, wherein the recovery operation is performed during the initializing the flash memory.

10. (previously presented): The apparatus as claimed in claim 5, wherein the recovering from the error is performed during reclaiming the flash memory wherein the reclaiming comprises moving data written in a predetermined unit of the flash memory to a new unit.

11. (currently amended): A flash memory access method comprising:
accessing the flash memory and searching for a currently writable physical block if a processor requests a write operation for a specific logical block number of the flash memory; and

writing data and meta-information in a physical block corresponding to a logical block with the logical block number if a previous write operation has not been performed for the logical block, and writing the data and the meta-information in a new physical block corresponding to the logical block without changing flash memory state information, wherein the flash memory state information is written in a previous physical block corresponding to the logical block if the previous write operation has been performed for the logical block, wherein the flash memory state information is time independent.

12. (original): The method as claimed in claim 11, wherein the meta-information is written after the data of the logical block is written.

13. (original): The method as claimed in claim 11, wherein the data and meta-information of the logical block are simultaneously written.

14. (previously presented): The method as claimed in claim 11, wherein the meta-information comprises the logical block number, and the flash memory state information indicating a state of the physical block as valid, deleted, or invalid.

15. (previously presented): The method as claimed in claim 11, further comprising a recovery operation comprising detecting, during a scanning process, physical blocks for the

logical block number and of recovering from an error by determining a valid block for the logical block among the detected physical blocks.

16. (previously presented): The method as claimed in claim 15, wherein the scanning process comprises:

reading a logical block number for each of the physical blocks by investigating the flash memory based on a latest accessed block, and investigating a field of a block allocation table corresponding to the read logical block number; and

writing a state value of "1" in the field of the block allocation table if the state value has been "0," and detecting that the logical block number has been searched for through the previous physical block during the scanning process, if the state value is "1."

17. (previously presented): The method as claimed in claim 15, wherein the recovering comprises recovering from the error by determining a latest data written among data of a specific logical block number detected during reclaiming the flash memory, and wherein the reclaiming comprises moving data written in a predetermined unit of the flash memory to a new unit.

18. (currently amended): A flash memory access apparatus, comprising:

a flash memory comprising a plurality of physical blocks, wherein each physical block of the plurality of physical blocks comprises data and meta-information, and the meta-information

~~comprises~~ a logical block number and flash memory state information ~~comprising data and meta-information~~, wherein the flash memory state information is time independent; and

a flash memory controller, wherein if a write operation is requested for the logical block number, the flash memory controller performs one of:

(a) a first write operation which writes the data and the meta-information in a first physical block corresponding to the logical block number, if the first write operation has not previously been performed for the logical block number, and changes the flash memory state information of the physical block corresponding to the logical block number, and

(b) a second write operation which writes the data and the meta-information ~~of in~~ a second physical block, if the first write operation has been performed for the logical block with the logical block number, wherein the flash memory state information of the physical block corresponding to the logical block number is not changed.

19. (previously presented): The apparatus as claimed in claim 18, wherein each physical block comprises further comprises a main area which stores the data and a spare area which stores the meta-information, the logical block number and the flash memory state information.

20. (previously presented): The apparatus as claimed in claim 19, wherein the meta-information is written in the spare area after the data of the logical block is written in the main area.

21. (previously presented): The apparatus as claimed in claim 19, wherein the meta-information is written in the spare area simultaneously as the data of the logical block is written in the main area.